Sphere Packing Aided Surface Reconstruction for Multi-view Data

Kun Liu, Rhaleb Zayer, Patricio Alejandro Galindo

Sphere Packing Aided Surface Reconstruction for Multi-View Data

Kun Liu, Rhaleb Zayer and Patricio Galindo

INRIA, Nancy-France
kun.liu@inria.fr

Abstract. Surface reconstruction has long been targeted at scan data. With the rise of multi-view acquisition, existing surface reconstruction techniques often turn out to be ill adapted to the highly irregular sampling and multilayered aspect of such data. In this paper, a novel surface reconstruction technique is developed to address these new challenges by means of an advancing front guided by a sphere packing methodology. The method is fairly simple and can efficiently triangulate point clouds into high quality meshes. The substantiated experimental results demonstrate the robustness and the generality of the proposed method.

1 Introduction

Reconstructing a surface from a point cloud is a well studied problem in geometry processing [1,2,3]. Historically, these methods have been tailored for range scan data. The rapid development of technologies for point cloud acquisition raises new challenges as the point data obtained by different approaches exhibit distinct properties in terms of density, accuracy, distribution, and so forth. Moreover, additional post-processing operations such as mesh smoothing [4,5], remeshing [6,7] and mesh simplification [8] are often necessary to refine the reconstructed surfaces and make them exploitable in practice. Therefore, it is still challenging to design an all-purpose surface reconstruction algorithm.

In this paper, a novel method for surface reconstruction is presented with special attention to point clouds generated by multi-view stereo reconstruction. An ideal surface reconstruction algorithm is characterized by several properties such as robustness to noise, low computational cost, and high quality of resulting meshes. However, these attributes are often hard to combine within a single approach. In fact, many existing methods generally suffer from some limitations. As discussed in the literature survey on surface reconstruction [9], for instance, classical Delaunay-based methods [3,10] cannot directly handle noisy data, implicit surface methods [1,2] inevitably create skinny triangles, and the efficiency of learning based methods remains a significant problem.

Awareness of the particular aspects of multi-view data has prompted recent interest in filling the gap left by classical geometry reconstruction methods. For instance [11] relies on a combination of restricted Voronoi diagrams and depth maps, whereas methods such as [12] recast the reconstruction problem as the

recovery of a visibility consistent surface from an initial Delaunay triangulation. This problem is addressed as finding a minimum s-t cut over an adaptive domain. Although these approaches can achieve highly accurate results the algorithms are rather intricate and time consuming.

The method proposed in this paper operates locally by means of an *advancing front* strategy guided by a global criterion based on *sphere packing*. Sphere packing is well known to approximate Voronoi diagram commonly used for producing high quality meshes [13]. In this way, robustness to noise in the input data and efficiency are achieved while keeping algorithmic approach simple. Noticeably, the method uses only points from the original point cloud as the vertices of the resulting mesh (subject to small perturbation possibly). This feature can benefit several applications such as tracking in computer vision and photogrammetry as it reduces the additional re-projection errors across frames. The proposed approach is evaluated both on multi-view stereo and scanning data. The results also demonstrate the efficiency and the robustness of the proposed approach.

The rest of this paper is organized as follows: previous work on sphere packing and advancing front are reviewed in Section 2. The reconstruction algorithm is presented in detail in Section 3, and experimental results are discussed in Section 4.

2 Related work

To keep this exposition succinct, we will restrict ourselves to the literature related to the methods underlying our approach, namely sphere packing and advancing front. For an extensive review of classical surface reconstruction techniques the reader is referred to [14,9].

2.1 Sphere packing

Sphere packing is a classical problem in geometry. It seeks an arrangement of non-overlapping spheres in a given metric space [15]. Sphere packing has been used to generate meshes for finite element analysis under the bubble mesh analogy [13]. The basic idea of the bubble mesh algorithm in [13] consists of packing spheres tightly within a predefined domain. In general, the pattern of the tightly packed spheres mimics a *Voronoi diagram* and thus the generated triangulation can be regarded as an approximation of a *Delaunay triangulation*. This property results in well-shaped triangular meshes whose elements are approximately equilateral, which can benefit many applications especially finite element analysis [16]. Sphere packing importance has been also recognized in the context architectural freeform designs where it is used to generate the so called *circle packing meshes* [17]. A circle packing mesh is a triangular mesh approximating an arbitrary freeform shape. The incircles of triangles in the mesh form a circle packing and the associated orthogonal spheres centering in vertices form a sphere packing. Since the resulting mesh is a torsion-free support structure, it exhibits remarkable aesthetic properties. To generate such meshes, an initial mesh

is computed and then morphed to a circle packing mesh which also maintains a close approximation of the input freeform shape.

Both the bubble mesh and the circle packing mesh methods can generate a triangulation for an input domain which possibly represents a CAD model or an architectural surface. The input domain is usually defined as an algebraic form. On the other hand, in this paper a point cloud, which is a type of discrete geometry representation, is triangulated. Hence, intermediate interpolation is necessary during surface reconstruction in order to predict the absent geometry, which makes the problem more challenging. Furthermore, the proposed approach does not apply any tedious global optimization as in [13,17], but still can produce meshes with satisfactory quality.

2.2 Advancing front

The advancing front strategy is widely used in mesh generation and surface reconstruction. The Ball-Pivoting method [18] adopts this strategy to triangulate a point cloud into a mesh which is a subset of alpha shape [10] of the input points. Therefore, geometric and topological correctness are guaranteed theoretically under certain sampling conditions. However, these conditions are too rigorous in practice. The basic idea of the surface reconstruction methods using advancing front is propagating, i.e., growing the reconstructed mesh from a seed triangle until certain termination conditions are satisfied. In [19], a guidance field based on curvatures is introduced to direct the propagation and an adaptive mesh is generated with the bounded reconstruction error. In addition, due to moving least squares (MLS) used for the interpolation in the method, noise in point data can also be handled.

Advancing front can benefit surface reconstruction in several aspects. Since distant triangles are created independently, parallel computing can be easily employed as demonstrated in the parallel Ball-Pivoting method [20]. Moreover, as the advancing front algorithms only process points locally, it is not necessary to load the whole point cloud into the memory simultaneously. This facilitates triangulating huge point clouds by means of streaming techniques. In this paper, advancing front is also adopted and sphere packing is used to direct the propagation during the reconstruction. To use the strategy of advancing front, three issues need to be carefully addressed, i.e., initialization, propagation and termination condition, which is comprehensively discussed in subsection 3.2.

3 Our method

In this section, we outline the main steps of our approach and show how sphere packing can be used to drive the advancing front in order to triangulate a point cloud into a mesh.

3.1 Sphere candidates generation

In order to deal with the irregular sampling commonly encountered in multi-view data, we first seek to associate a sphere radius r_i with every point v_i in the data set in a similar way to the work of [21]. For this purpose, a normal direction for each point of the cloud is estimated using *principal component analysis* [22]. In our method, oriented normals are not necessary as the normal direction is sufficient. With the computed normal direction N_i on the point v_i, a local plane is determined and the sphere candidate centered at v_i is defined as the largest sphere such that points in the sphere are all less than a prescribed ϵ distant from the plane. Therefore, the parameter ϵ can be used to bound the reconstruction error. The ϵ also can be considered as a threshold on the size of resulting meshes [21].

3.2 Advancing front with packing criterion

In order to build a triangulation based on the candidate spheres obtained in the subsection 3.1, we develop an advancing front strategy where propagation is steered by sphere packing.

Typically advancing front methods are characterized by three main aspects, namely, **i**) **front initialization**, **ii**) **propagation rules** and **iii**) **termination conditions**. Among these, the key aspect is propagation rules where the principle which drives the front expansion is defined, e.g., pivoting ball as in [18] or curvature field as in [19]. In our proposed method, a criterion based on sphere packing is developed to guide the propagation - growing from a seed triangle to a triangular mesh covering the whole surface. The algorithmic outline of the proposed method is described in Algorithm 1.

i) Front initialization The proposed advancing front algorithm starts from a seed triangle, and the edges of the seed triangle form the initial fronts. To generate a seed triangle, a sphere (v_1, r_1) is chosen randomly, where the sphere centered in v_1 with radius r_1. The second sphere (v_2, r_2) is selected using the following criterion

$$\min_{(v_2,r_2)} PairError((v_1, r_1), (v_2, r_2)), \tag{1}$$

where $PairError((v_1, r_1), (v_2, r_2)) = |r_1 + r_2 - |v_1 - v_2||$. The third sphere (v_3, r_3) is chosen as

$$\min_{(v_3,r_3)} \max\{PairError((v_1, r_1), (v_3, r_3)), PairError((v_2, r_2), (v_3, r_3))\}. \tag{2}$$

After adjusting the radii to make the three spheres tangent to each other, the seed triangle with the vertices v_1, v_2 and v_3 is determined, and thus the fronts are initialized as the edges (v_1, v_2), (v_2, v_3) and (v_3, v_1).

Algorithm 1 Advancing front in our method

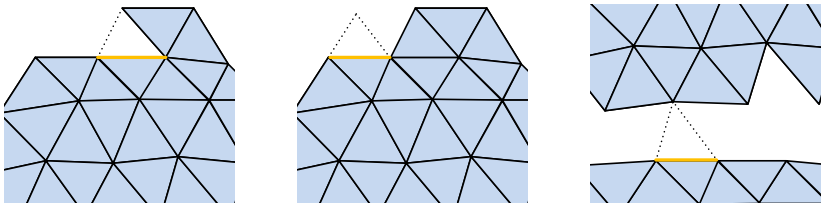
Input: point cloud p, normals n, spheres s
Output: triangle list *tri_list*

```
fronts ← FRONTS_INITIALIZATION()                ▷ See front initialization
tri_list ← ∅
while fronts.size() > 0 do
    e ← fronts.pop()
    t ← EAR_CUTTING(e)                          ▷ See propagation rules
    if t ≠ ∅ then
        tri_list.push_back(t)
        fronts.update()
        continue
    end if
    t ← ADDING_POINT(e)                         ▷ See propagation rules
    if t ≠ ∅ then
        tri_list.push_back(t)
        fronts.update()
        continue
    end if
    t ← MERGING_FRONTS(e)                       ▷ See propagation rules
    if t ≠ ∅ then
        tri_list.push_back(t)
        fronts.update()
        continue
    end if
end while
```

ii) Propagation rules Once the fronts are initialized, a series of operations are sequentially applied to the current front until a new triangle is successfully created. These operations include *ear cutting*, *point addition*, and *merging fronts* as described in Algorithm 1. The specific rules related to each operation will be briefly described below.

Ear cutting: This operation is applied to check if the current front edge can create a new triangle with an adjacent edge as illustrated in Figure 1a. Before creating the triangle, a set of four *filters* discussed in subsection 3.3 need to be passed. In this way, unsatisfactory triangle candidates are eliminated, e.g., non-manifold triangles, self-intersected triangles and skinny triangles.

point addition: If a new triangle cannot be created after applying the ear cutting operation, the point addition operation is consequentially explored. This operation creates a few triangle candidates by using the current front and its nearby points as shown in Figure 1b. Again in this case, the four filters in subsection 3.3 are used before new triangles are validated. If none of the candidates can pass the filters, the next operation, i.e., merging fronts, is performed. Otherwise, the created triangle is chosen from the candidates using Equation 2. In this way, a tight packing configuration is guaranteed. Once an optimal triangle is found,



(a) (b) (c)

Fig. 1: Three operations applied for advancing the current front (yellow): (a) ear cutting; (b) point addition; (c) merging fronts.

the spheres centered at the three vertices are shrunk to prevent sphere intersections. In our implementation, a local quadratic surface is constructed [19] before applying the point addition operation. The use of this local surface allows the elimination of noise in the data.

Merging fronts: This operation is used to merge different fronts as illustrated in Figure 1c. In principle, it is similar to the point addition operation except two distinctions. First, it creates triangles only using points along fronts. Second, no new points (spheres) are introduced and thus shrinking the spheres is not necessary.

iii) Termination conditions In the implementation, a queue data structure is utilized to manage front information. Once a triangle is created, the current front is removed from the queue and the edges of the new triangle are marked as fronts, Otherwise, only the current front edge is deleted. The propagation is terminated when the queue is empty. Furthermore, the queue is a First-In-First-Out (FIFO) data structure. It leads to an approximate radial propagation which prevents ubiquitous indentations along fronts.

3.3 Filters in front propagation

Four filters are utilized throughout propagation operations to control the validity of new triangle candidates.

a) non-manifold edge filter: checks if new triangles may introduce non-manifold edges.

b) triangle orientation filter: prevents orientation flipping. In our experiments, the angle of two adjacent triangles is required to be lower than 90 degrees.

c) the triangle overlapping filter: helps avoid spatial overlapping of nearby triangles. To detect the overlapping of two triangles, one triangle is projected to the plane determined by the other one and vice versa. The triangles are spatially overlapped if the projected triangles have intersections in the plane.

d) the triangle quality filter: guarantees the quality of triangles by using constraints on triangle angles. Different constraints are used in the different

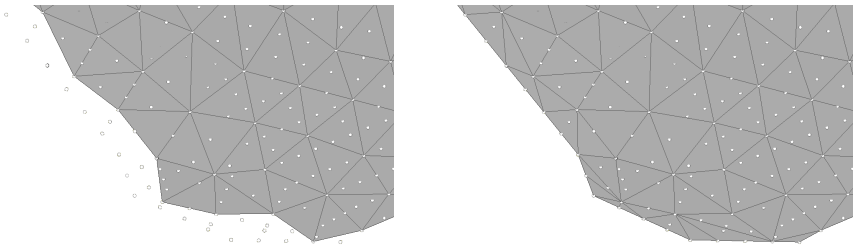
operations. In the ear cutting operation, the angle between the current front and the adjacent edge is required to be less than 90 degrees. In the point addition operation, the minimum angle in the new triangle is bounded by a threshold. In our experiments, 30 degrees is prescribed. In the merging fonts operation, the two angles associated with the current front are required be acute.

3.4 Refinement

Once a new triangle has been created during advancing front, an edge swapping operation can be applied if the edge length decreases by the swapping [23]. This operation helps improve the quality of the resulting mesh.

Boundary treatment: Once the mesh is constructed by the advancing front method presented in subsection 3.2, a supplementary propagation can be employed to refine boundary regions illustrated in Figure 2a. In principle, this refinement operates similarly to the propagation in subsection 3.2 except that it uses weaker triangle quality filters, i.e., with relaxed parameters. By applying the supplementary propagation, the results can be significantly improved as shown in Figure 2b.

Hole filling: Due to the nature of multi-view data, often there are small regions where data can be missing (for instance due to the lack of matches across views). For such cases, we use the hole filling algorithm [24] in post processing step. Although the algorithm has a $O(n^3)$ time complexity, most of holes are small and thus it operates reasonably. In our experiments, the algorithm is automatically applied to small holes with less than 50 edges.



(a) (b)

Fig. 2: The comparison of (a) without and (b) with the supplementary propagation. The white dots represent the input point cloud.

4 Results

The proposed method is evaluated on point cloud data acquired by multi-view stereo reconstruction as well as laser scan data. For the multi-view setting we

performed tests on the benchmark proposed in [25]. The point cloud data is obtained based on the quasi-dense matching approach of [26]. Only point coordinates are given as input. Our proposed method can handle the orientation problem as stated in subsection 3.1. The reconstructed meshes are shown for the fountain-P11 and the Herz-Jesu data sets in Figure 3.

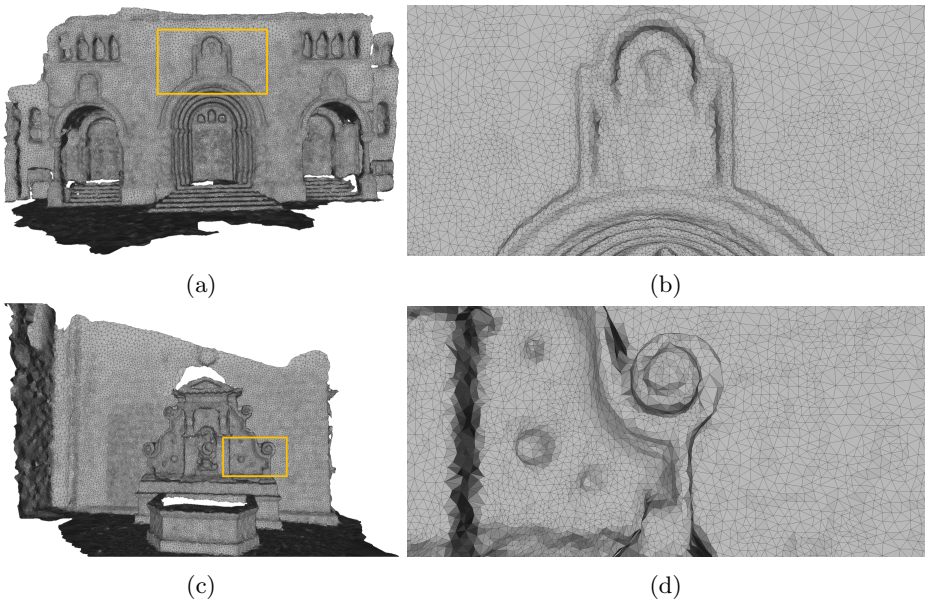
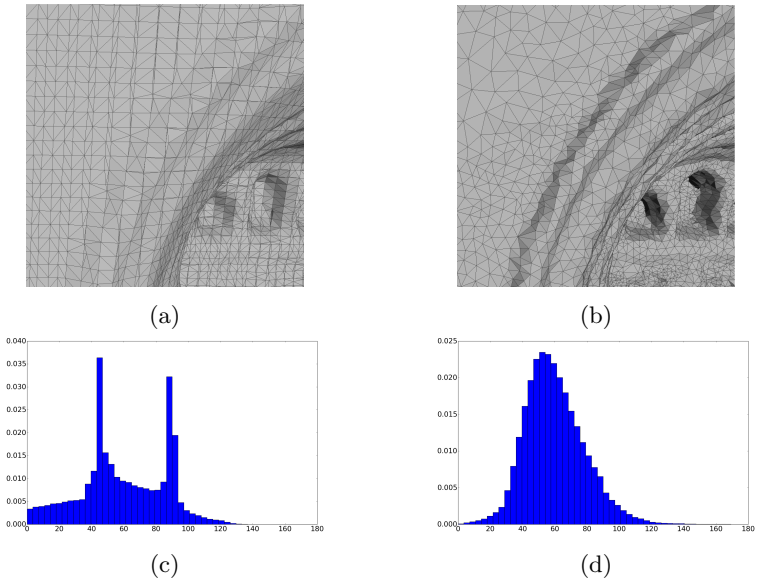


Fig. 3: Two point clouds corresponding to multi-view data from [25] are triangulated using our proposed method. (a) and (c) display the two resulting meshes. (b) and (d) are the corresponding close-up views.

Our results are compared to the Poisson reconstruction method [2] as it is one of the most popular surface reconstruction methods especially for multi-view stereo data. A zoom on the reconstructed Herz-Jesu, Figure 4, reveals a considerable number of skinny triangles resulting from the Poisson reconstruction, Figure 4a, whereas the triangles generated by our method are generally are well shaped, Figure 4b. This is confirmed by the histogram representation of triangle angle values in Figure 4c and Figure 4d.

The corresponding Table 1 lists detailed statistics about angles. The resulting Poisson reconstruction exhibits many angles lower than 30 degrees. However, most of angles in our resulting mesh are between 30 degrees and 90 degrees as well as within a smaller standard deviation, which indicates the generated triangles are close to equilateral triangles. It is considered as an important property of high-quality meshes [16].

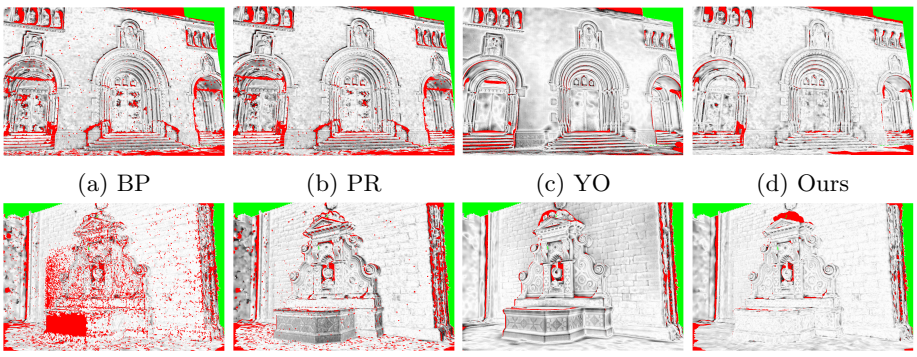


(a) (b) (c) (d)

Fig. 4: A comparison of Poisson reconstruction (a) and our proposed method (b) is illustrated. The input point cloud is same to the one used in in Figure 3a and the two meshes are displayed in the same close-up view. (c) and (d) are two corresponding histograms about triangle angle values.

	0°-30°	30°-60°	60°-90°	90°-120°	120°-150°	150°-180°	standard deviation
Figure 4a	13.05%	40.06%	33.54%	12.80%	0.54%	0.01%	26.50°
Figure 4b	2.91%	51.66%	39.16%	5.76%	0.43%	0.08%	18.76°

Table 1: The statistics about the angles in the results Figure 4a and Figure 4b.



(a) BP (b) PR (c) YO (d) Ours

(e) BP (f) PR (g) YO (h) Ours

Fig. 5: The reconstruction results of BP, PR and YO in [18,21,2] respectively, as well as our methods are evaluated using the method proposed in [25]. The images show the variance weighted depth difference. Red pixels represent errors larger than 30σ. Green pixels represent the missing scan data of the ground truth. The relative errors between 0 and 30σ are displayed using gray scale from 255 to 0.

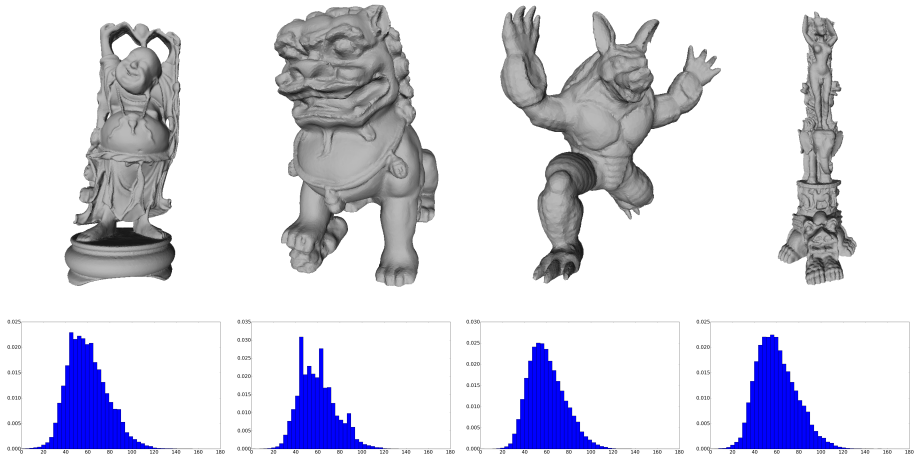


Fig. 6: The resulting meshes of our method on scan point data: The first row shows the resulting meshes; The second row shows the corresponding histograms about triangle angles in the meshes.

During the meshing process, most of existing methods methods such as Poisson reconstruction [2] introduce additional points into resulting meshes. Our proposed approach is different - only points from the input point cloud are used as vertices of the resulting mesh. This property prevents additional errors caused by interpolation, which can benefit many applications such as tracking in computer vision and photogrammetry. This is demonstrated by the evaluation benchmark proposed in [25] where ground truth data (from a laser scan) can be used to evaluate the reconstruction error. We compare our approach to different mesh reconstruction techniques, namely, the Ball-Pivoting [18], the method in [21], Poisson reconstruction [2]. The results are summarized in Figure 5. Brighter color corresponds to lower error w.r.t ground truth.

In general, our method performs relatively well and yields fewer red pixels. which suggests our results are closer to the ground truth. Detailed evaluation of the corresponding relative errors and completeness is summarized in Table 2. As a sanity check, we further performed reconstruction on point cloud data with added Gaussian synthetic noise. The reconstruction results are shown in Figure 6 as well as the histograms about triangle angles, which demonstrates the robustness and the generality of our method. The sizes of point clouds and resulting meshes in our experiments are also listed in Table 3.

5 Conclusion

In this paper, a novel surface reconstruction method is proposed based on intertwining sphere packing with an advancing front strategy. The local nature of the advancing front and the flexibility of sphere packing allow addressing the highly

	Fig.5a	Fig.5b	Fig.5c	Fig.5d	Fig.5e	Fig.5f	Fig.5g	Fig.5h
Completeness	75.83%	78.83%	76.79%	86.38%	73.51%	82.79%	79.96%	91.74%
Relative error	2.87	3.14	2.99	2.91	2.22	3.33	2.89	0.88

Table 2: The completeness and errors of the results in Figure 5.

Model	# of points	# of vertices	# of triangles
Fig.3a	630057	54126	106057
Fig.3c	359840	39470	77957
Fig.6 left	543652	92689	185236
Fig.6 middle left	655980	88031	176058
Fig.6 middle right	196269	26860	49716
Fig.6 right	180754	24837	48117

Table 3: The sizes of the input point clouds and the resulting meshes in Section 4.

irregular input commonly encountered in multi-view data. The resulting meshes are well-shaped and approximate well the original geometry.

Theoretical guarantees in terms of geometric and topological correctness in the light of the work [10] will be explored in future work.

References

1. Lorensen, W.E., Cline, H.E.: Marching cubes: A high resolution 3d surface construction algorithm. In: ACM Siggraph Computer Graphics. Volume 21., ACM (1987) 163–169
2. Kazhdan, M., Bolitho, M., Hoppe, H.: Poisson surface reconstruction. In: Proceedings of the fourth Eurographics symposium on Geometry processing. SGP '06, Eurographics Association (2006) 61–70
3. Dey, T.K.: Curve and surface reconstruction: algorithms with mathematical analysis. Volume 23. Cambridge University Press (2007)
4. Desbrun, M., Meyer, M., Schröder, P., Barr, A.: Implicit fairing of irregular meshes using diffusion and curvature flow. In: Proceedings of the 26th annual conference on Computer graphics and interactive techniques. (1999) 317–324
5. Fleishman, S., Drori, I., Cohen-Or, D.: Bilateral mesh denoising. In: ACM Transactions on Graphics (TOG). Volume 22., ACM (2003) 950–953
6. Yan, D.M., Lévy, B., Liu, Y., Sun, F., Wang, W.: Isotropic remeshing with fast and exact computation of restricted voronoi diagram. In: Computer graphics forum. Volume 28., Wiley Online Library (2009) 1445–1454
7. Attene, M., Falcidieno, B.: Remesh: An interactive environment to edit and repair triangle meshes. In: Shape Modeling and Applications, 2006. SMI 2006. IEEE International Conference on, IEEE (2006) 41–41
8. Heckbert, P.S., Garland, M.: Optimal triangulation and quadric-based surface simplification. Comput. Geom. Theory Appl. **14** (1999) 49–65
9. Schall, O., Samozino, M.: Surface from Scattered Points: A Brief Survey of Recent Developments. In Falcidieno, B., Magnenat-Thalmann, N., eds.: 1st Inter-

national Workshop on Semantic Virtual Environments, Villars, Switzerland, MIRALab (2005) 138–147
10. Edelsbrunner, H., Mücke, E.P.: Three-dimensional alpha shapes. ACM Transactions on Graphics (TOG) **13** (1994) 43–72
11. Salman, N., Yvinec, M.: Surface Reconstruction from Multi-View Stereo. Lecture notes in computer science (2009)
12. Vu, H.H., Labatut, P., Pons, J.P., Keriven, R.: High accuracy and visibility-consistent dense multiview stereo. Pattern Analysis and Machine Intelligence, IEEE Transactions on **34** (2012) 889–901
13. Shimada, K., Gossard, D.C.: Bubble mesh: automated triangular meshing of non-manifold geometry by sphere packing. In: Proceedings of the third ACM symposium on Solid modeling and applications. SMA '95, New York, NY, USA, ACM (1995) 409–419
14. AIM@SHAPE: Survey acquisition and reconstruction. Technical report (2004)
15. Sloane, N.J., Conway, J., et al.: Sphere packings, lattices and groups. Volume 290. Springer (1999)
16. Dey, T.K., Bajaj, C.L., Sugihara, K.: On good triangulations in three dimensions. International Journal of Computational Geometry & Applications **2** (1992) 75–95
17. Schiftner, A., Höbinger, M., Wallner, J., Pottmann, H.: Packing circles and spheres on surfaces. ACM Trans. Graph. **28** (2009) 139:1–139:8
18. Bernardini, F., Mittleman, J., Rushmeier, H., Silva, C., Taubin, G.: The ball-pivoting algorithm for surface reconstruction. Visualization and Computer Graphics, IEEE Transactions on **5** (1999) 349–359
19. Scheidegger, C.E., Fleishman, S., Silva, C.T.: Triangulating point set surfaces with bounded error. In: Proceedings of the third Eurographics symposium on Geometry processing. SGP '05, Aire-la-Ville, Switzerland, Switzerland, Eurographics Association (2005)
20. Digne, J.: An analysis and implementation of a parallel ball pivoting algorithm. IPOL (2013) 1–17 accepted for publication - to appear.
21. Ohtake, Y., Belyaev, A., Seidel, H.P.: An integrating approach to meshing scattered point data. In: Proceedings of the 2005 ACM symposium on Solid and physical modeling. SPM '05, New York, NY, USA, ACM (2005) 61–69
22. Hoppe, H., DeRose, T., Duchamp, T., McDonald, J., Stuetzle, W.: Surface reconstruction from unorganized points. SIGGRAPH Comput. Graph. **26** (1992) 71–78
23. Attene, M., Falcidieno, B.: Remesh: An interactive environment to edit and repair triangle meshes. In: Proceedings of the IEEE International Conference on Shape Modeling and Applications 2006. SMI '06, Washington, DC, USA, IEEE Computer Society (2006) 41–
24. Liepa, P.: Filling holes in meshes. In: Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing. SGP '03, Aire-la-Ville, Switzerland, Switzerland, Eurographics Association (2003) 200–205
25. Strecha, C., von Hansen, W., Van Gool, L., Fua, P., Thoennessen, U.: On benchmarking camera calibration and multi-view stereo for high resolution imagery. In: Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, IEEE (2008) 1–8
26. Kannala, J., Brandt, S.S.: Quasi-dense wide baseline matching using match propagation. In: Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on, IEEE (2007) 1–8